書いた人：xor

あなたのコンピュータの中に入ってる大事な情報は何ですか？ 会社の極秘書類か、それとも SSH の秘密鍵か、あるいは $n(\in \mathbb{N})$ 年かけて溜めたいかがわしい画像や動画か？ まぁ、それが何であるかは俺の知ったことじゃないのですが、問題はその保存方法。長らくコンピュータを使っている人間ならば痛感していることだと思うんですが、コンピュータに保存された情報は、常に様々な危険にさらされているのです。何らかの対策を施さなければ、あなたの大事な情報はある日突然夢の中へと消えてしまうでしょう。で、いざそうなったときに「トラブルを起こす PC がいけないんだ！」と責任転嫁されても困るので、とりあえず「トラブルが起きないようにする and 起きたときにも簡単に復旧できるようにする」方法を調べてまとめてみました。

1　経緯

ここだけの話なんですが、本来ここには「俺と情報化社会」という題名の、40 ページにも及ぶ壮大なエッセイが載るはずだったんですよ。実際、俺は 37 ページ目まで書きあげたし、ここまで書いておいて完成しない方がおかしいっすよ〜HAHAHA、というところまで来ていたんです。

それにも関わらず、こんなヘッポコタイトルの記事になってしまったのは何故なのか……。俺の壮大なエッセイは、ルートディレクトリにて誤って実行した `rm -fr *` によって跡形もなく消え去ってしまったのです。Windows でいうと、今までの作業を全てゴミ箱に移し、移動が終了した直後にゴミ箱の中身を空っぽにしたようなもんでしょうか。ホームディレクトリで 3 回くらい ls し、以後何回 ls してもファイル名は出てこないんだろうなと悟ったときは、流石負け組だぜ！ と思ったものです。アーッ！ どう頑張っても、俺のエッセイは戻ってこない。俺の数日間の努力は全部無駄になってしまったアアアアアア！

仕方がないので新しく書き直したこの原稿なんですが、これも一時は触ることすらできなくなりました。上に書いた惨事のあとで、今度は作業用の PC が壊れてしまったのです。突然 HDD がの死に、後を追うようにメモリが死に、結局マザーボードと CPU と HDD を新調する羽目に……。幸いデータ用 HDD は無事で、中に入っていたデータも救出することができたので、今こうして皆さんの前に文章を披露することができるわけですが……あのとき、壊れていたのがデータ用 HDD だったら、と考えると背中に寒気が走ります。

残念なことに、大事な情報を完璧に護る手段はありません。しかし、せっかく作ったデータが救出不能になるのを予防する方法はいろいろあります。それを少しでも多く伝えることができれば、それ以上のことはありません。

2　脅威

ここでは、大事な情報が消滅してしまう原因となりうる事柄を「脅威」と呼ぶことにします。情報セキュリティの分野では、情報資産の脅威を物理的脅威・技術的脅威・人的脅威と分類しています[1]。情報セキュリティでは第三者による情報漏洩なども脅威に含まれるんですが、ここではそれらを除外いたします。

2.1　物理的脅威

おそらく、もっともメジャーな脅威かと思われます。つまるところ、データが保存してある媒体に対して何らかの物理的攻撃がなされることを言うのですが、具体的なものには次のようなものがあります。

- 地震・洪水・津波・火災などの自然災害
- 停電・雷サージ[2]などの電源関連の事故
- 物理的に家に侵入、ハードディスクをフルボッコ

また、ハードディスクは消耗品なので、仮に攻撃されなくても寿命が来たら壊れてしまいます。これはハードディスクに限らず、全ての物理的な媒体にも言えること。CD や DVD も例外ではありません。

2.2　技術的脅威

特に個人のコンピュータで発生しうる技術的脅威の代表は、コンピュータウィルスによるものでしょう。コンピュータウィルスの中には、大事な情報を削除や改竄してしまうものもあります。最近のウィルスは非常に巧妙にできていますから、油断しているとあっという間に感染してしまうでしょう[3]。

個人でサーバを立てているなら、不正アクセスも技術的脅威となりえます。その中に大事な情報が入っているのなら、それなりの対策も必要になるでしょうね。

なお、作業中のファイルを保存しないままアプリケーションが不正終了してしまった場合についても、この脅威に分類いたします。私の独断と偏見によって。だってねぇ、アプリケーションが不正終了するって、どう考えても技術不足によるものでしょう。

2.3　人的脅威

人的脅威とは何かと申しますと、冒頭に書いたような、コンピュータを操作する人間によって引き起こされる脅威のことです。情報セキュリティの観点からいうと、上に示した誤操作以外にも

- 外部へのデータの持ち出し
- パスワードの不正利用

なども挙げられるんですが、個人的には、自分しか使わないコンピュータにおける人的脅威は誤操作くらいしかないんじゃないかと思っています。

誤操作なんて、って馬鹿にはできませんよ。どんなに高価な装置を使っていても、こればかりは避けられませんからね。そんなの、注意して操作しない方が悪いんだよ、とおっしゃる方もいらっしゃるかもしれません。ですが、悲しいことに、人間というのは過ちを犯す生物なのです。誰もが注意して操作していればミスは発生しないだろう、なんて考えている人は、その考え自体がミスであることに気づいてください。そうじゃない、そうじゃないんですよ……。

3　対策

上に述べた脅威を見てどう思うかは人それぞれでしょうね。中には「俺のところに限って……」と鼻で笑う人がいるかもしれません。ですが、マーフィーの法則にあるように、「失敗する可能性のあることは、必ず失敗する」のです！ 失敗してからでは遅すぎます。

対策方法は、大まかに分けてフールプルーフ (Fool proof:失敗しづらくする)、バックアップ (Backup:失敗しても以前の状態に戻せる)、冗長構成 (同じものを複数用意する) などがあります。それぞれについて簡単に説明いたします。

[1]筆者は、この会誌が皆さんの手に届く前に、情報セキュリティアドミニストレータの試験を受ける予定である。受験するのは 2 回目なのに、どうも前回よりも悪い結果を出すような気がするのは何故なんだろうか……。

[2]雷が落ちることによる電圧上昇のこと。

[3]筆者の PC も去年、「WinFixer」「CoolWebSearch」というウィルス (正確にはスパイウェア) に感染してしまった。特に WinFixer は、ご丁寧に HTTPS のデジタル証明書までくっついてきていた。その時のルート CA は Thawte Code Signing CA であった。それ以後、ルート CA が Thawte Code Signing CA であるデジタル証明書は一切受け付けないことにしている。

3.1 フールプルーフ

コマンドラインを操作しているうちに大事な情報を間違って削除してしまう可能性について考えてみましょう。そもそもどうして間違って大事なデータを削除してしまうのか？ 判断ミスもあるでしょうが、一番の問題は**非常に簡単に大事なデータが消せてしまう**というところにあると思います。

データを簡単に消せなくすることは非常に有効な手段です。たとえばWindowsの場合、データを削除するためには

1. まずデータに対して「削除」を実行
2. すると「削除しますか？」と訊かれるので、はいと答える
3. さらに「ゴミ箱を空にする」を実行
4. すると「本当に空にしますか？」と訊かれるので、はいと答える

の4段階を踏む必要があります。最後の段階を除いて、いつでも引き返せます。それにひきかえ、UNIXでは

1. `rm` ファイル名と打ち込んでEnterを叩く

しかありません。引き返す間もなくデータは消滅いたします。不慮のミスだからって容赦はいたしません。

……これではミスを防ぐことはできませんね。

3.1.1 rmの削除警告表示

`rm`コマンドには、`i`というオプションがあります。同じファイル名を削除するのにも、`rm -i` ファイル名と打ち込むことによって、削除対象のファイル1個1個に対して本当に削除するかを問い合わせるようになります。これでファイルを削除するまでに必要な工数が1増えるので、重要なファイルを削除するハードルがちょっとだけ高くなります。

なお、いちいち`-i`と打ち込むのはだるいので、bashを使っているのならば`.bashrc`、cshを使っているのならば`.cshrc`などの設定ファイルに

```
alias rm='rm -i'
```

と書き加えておくとよいでしょう。こうすることによって、`rm`と打ち込むことと`rm -i`と打ち込むことが同じになります[4][5]

3.1.2 疑似ゴミ箱作成

さらに安全なのは、Windowsでいうゴミ箱を作ることです。最初にゴミ箱代わりのディレクトリを作っておき、`rm`コマンドが入力されたら、削除せずにそのディレクトリに移動するだけです。そのためには、まず次のようなシェルスクリプトを用意します。

```
#!/bin/sh
if[! -d ~/trash] then
  mkdir -p ~/trash
fi
for i do
  mv $i ~/trash
done
```

このファイルを`~/bin/mvt.sh`に保存したとしましょう。つぎに、このファイルに

```
chmod +x ~/bin/mvt.sh
```

とすることで、実行権限を持たます。最後に、`.bashrc`などの設定ファイルに

```
alias rm='~/bin/mvt.sh'
```

と書き、自分のホームディレクトリに`trash`というディレクトリを作れば完成！ その際、前章に書いた`alias rm='rm -i'`は削除してくださいね。

この方法の優秀な点は、たとえばゴミ箱(`trash`ディレクトリ)の中身を空っぽにしようと思って`rm`とやっても、相変わらず`trash`ディレクトリに移動されるだけで、知識がないといつまでたっても削除できないことです。本当に削除するためにはあるテクニックを使う必要があるのですが、それはあえて書きません。……これにより、データを削除する際のハードルを高めることができた！ と言っても良いでしょう。

このゴミ箱スクリプトは最低限の機能しかついていないので、いろいろと不便するかもしれません。そんな

[4] ただし書き加えた直後はなされない！ 書き加えた直後にやるならその前に「source 設定ファイル名」を実行すること！

[5] なお、rm -i をすると、削除対象の全てのファイルに対して、削除を許可する場合には y、しない場合には n と入力する必要がある。それがめんどくさいという理由で作られた、ひたすら y(改行) と出力し続ける yes というコマンドが存在する。これを rm -i ファイル名 yes と使うことによって、i オプションをつけているにも関わらず何も訊かれずに全てのファイルを削除することができる。事故を防ぐためにも、yes コマンドは使用しないことをおすすめする。

ときは、Google にて「rm ゴミ箱」で検索すれば、もっと良いスクリプトを見つけることができるでしょう。

3.2 バックアップ

ご存じの方も多いと思われますが、一応説明しておきます。バックアップとは、通常とは別の場所にファイルのコピーを取っておくことです。こうすることで、たとえば誤って大事なデータを削除してしまったとしても、あらかじめ取っておいたコピーを元の場所に戻すだけで復元できます。これはかなり昔から使われてきた手法であります。

バックアップの最適な取り方は、環境によって異なります。ここでは、いくつかの項目に分けて解説します。

3.2.1 方法

通常、バックアップというのは何回も行うものです。しかし、バックアップする対象がそれなりに大きいと、どうやってバックアップするかも考えなければなりません。というのも、たとえばバックアップする対象が 512GB で、その全てに対して毎日バックアップを取ろうと思ったら、1ヶ月バックアップを取るのに 15TB というとんでもない領域が必要になってしまいます。たった 1ヶ月のために！

よほど荒い使い方をしない限り、1 日に 512GB のデータが全て編集されることはありません。編集されないファイルも存在するでしょう。そういうファイルは、前々日分のバックアップから復元できるのですから、新たにバックアップを取る必要はありません。そんなわけで、バックアップを取るときは前日のバックアップとの差分を取り、差分だけをバックアップする、という方法もあります。ただし、実際に障害が発生したら、バックアップを取りはじめてから障害が発生する直前までの全てのバックアップが必要になります。そのため復元 (リストア) 作業は大変、というかめんどくさいです。

そんなわけで、バックアップの方法としては主に以下の 3 つが用いられています。

フルバックアップ 全てのファイルをバックアップする方法。データの復元には、障害が発生する直前に取ったバックアップのみが必要です。

増分バックアップ 前回のバックアップ以降に更新されたデータのみをバックアップする方法。データの復元には、初回バックアップから障害発生直前までの全てのバックアップが必要です。

差分バックアップ バックアップのうち、何回に 1 回かは全てのファイルをバックアップ (FULL)、それ以外のときは、最終 FULL との差分のみをバックアップ (DIFF)、という方法。データの復元には、障害発生直前の最終 FULL と最終 DIFF のみが必要です。

バックアップの所要時間・復元の所要時間を考えると、我々一般大衆にもっとも向いているのは比較的バランスの良い差分バックアップだと俺は思います。

3.2.2 周期

特に規定はありません。会社などでは毎日バックアップするのが基本らしいのですが、個人の場合は……別に週 1 回でも月 2 回でも良いと思います。

ただし、周期が長くなる分、いざ障害が発生したときの損失も大きくなることは忘れないでくださいね。1 週間分のデータが消滅したときの精神的ダメージは相当でかいですよ〜。

3.2.3 媒体

バックアップに使えるかもしれない媒体を挙げてみました。それぞれについて説明いたしましょう[6]！

ハードディスク 磁性体を塗られた円盤と、それを読み取る「ヘッダ」からなる装置。販売されているものの容量は 40GB から 1TB、価格は 4500 円から 43000 円でした。書き換え回数は 100 万回、寿命は半年から 5 年[7]と言われています。

USB メモリ 内部は NAND 型フラッシュメモリと呼ばれるものです。最近になって浸透してきましたね。作者はアレが大っ嫌いですけど[8]。どうでもいいで

[6] なお、記述は全て、この文章を執筆している時点での情報である。

[7] しかし作者がデータ HDD として使っている 9.52GB の HDD は、6 年ほど回し続けているにも関わらず正常に動作している。

[8] 引っこ抜く度に「安全な取り外し」ってやらないといけないのがたまらなく堪らん。

すね。容量は 8MB から 64GB。価格は 128MB のものが 650 円[9]、64GB のものはおよそ 33 万円でした。書き換え回数は 10 万回ほど、寿命は 10 年ほどと言われています。

光ディスク バックアップに使える光ディスクとして挙げられるのは CD-R、記録可能 DVD 各種、Blu-ray Disk-R(以下 BD-R) や HD-DVD などでしょうか。最も記憶容量の少ない CD-R の容量は最大で 700MB、最も記憶容量の多い BD-R の容量は片面 4 層で 100GB[10]。CD-R の寿命はおよそ 30 年と言われています。

光磁気ディスク MO[11]ディスクと呼ばれるものです。容量は 128MB から 2.6GB。書き換え回数は 1000 万回、寿命は 50 年から 100 年と言われています[12]。

磁気テープ 実際はそうではないにも関わらず、古い・高い・遅い・等のイメージが定着してしまったかわいそうなデバイス。もちろん、そんなことはございません。企業などで使われているバックアップデバイスとしての磁気テープは、最新の技術の塊でございます。それこそ、ハードディスクなどの精密機械に匹敵するほどの技術が惜しみなく導入されているのです。また、実際の記憶媒体であるカートリッジの価格は 400GB のものが 1 万円ほどなので、GB 当たりの単価は圧倒的に安いです[13]。更に、ハードディスクなどは常時回しておく必要があるのに対して、テープは常に回しておく必要が無いので、電気代も大幅に削減できます。テープデバイスなのでアクセス速度 (HDD でいうシーク時間) は遅いですが、データ転送速度は HDD と遜色無いほど高速です。テープカートリッジの寿命はおよそ 30 年と言われています。

個人で使う範囲ならば、USB メモリか CD-R、あるいは HDD を使うのが良いと思われます。ここでは、HDD を使うと仮定して話を進めさせていただきます。

3.3 冗長構成

幸いなことに、重要なデータがメディアと共に死なずに、どこかに生き残らせることに成功したとします。ですが、データが生き残っていても、それを吸い出すことができなければ意味がありません。吸い出すことができないということは、死んでいることと同じ！ 待避しておいたデータを救済するまでが「対策」です。

データを救済することまで考慮に入れると、冗長化は欠かせないファクターです。冗長化とは、同じものを複数用意しておき、1 つが壊れても残りのを使えば大丈夫！ という構成にすることです。サーバやネットワーク、HDD やバックアップ装置などは、唯一しかない装置が壊れてしまうと再起不能な事態に陥ることが多いので、冗長化されることが多いですね。

ここでは、HDD の冗長化について説明します。HDD でバックアップを取るならばやっておきたい、RAID についてです。現段階で RAID には 0 から 7 があります。

3.3.1 RAID0

最低 2 台の HDD が必要。

例えば HDD が 2 台あって、記録するデータが 1,2,... となっているとき、片方の HDD には 1,3,5,...、もう片方には 2,4,6,... と記録します。この様子が縞模様に見えることから、この RAID はストライピングと呼ばれます。長所はもちろん、アクセス速度が HDD の台数に比例して高速になることです。

ですがコレ、厳密には RAID じゃありません。冗長性が全くないからです。っていうか、HDD の台数に比例して故障する確率が高くなるので、冗長性もへったくれも無い！ なんだコレは！

3.3.2 RAID1

最低 2 台の HDD が必要。

---

[9] 8MB のものの値段は見つけられなかった！

[10] まだ学会発表にとどまっているが。

[11] Magneto Optical

[12] この数字は誤植ではない。MO は、他のメディアに比べて耐久性に優れたメディアなのである。それは、他のメディアにおいて消耗の原因となる事柄が、MO にはほとんど通じないからである。具体的に述べると

- 加熱しないと磁気の書き換えができないので、磁石の近くに置いても影響を受けない
- 紫外線の影響をほとんど受けない
- ヘッドが接触しないので物理的磨耗が無い
- カートリッジに収まっているので、埃や疵の影響をうけにくい

などである。

[13] ただし、カートリッジを入れるデバイスの方は極めて高価である。Sun StorageTek Virtual Tape Library Plus というテープデバイスは、定価$135,000 である。

RAID に接続されている全ての HDD に、同じ内容を記録します。なので、1 台どころか同時に数台の HDD が壊れても、生きている HDD がある限り、データが完全に死ぬことはありません。

欠点は、HDD の容量に無駄が生じることと、同じことを複数の HDD に書くのでアクセス速度が低下するということです。また新しい HDD(中身空っぽ) を追加する際に、初期コピーの順番を間違えると、全ての HDD が新しい HDD と同じ内容になってしまう (つまりデータが全て無くなる) ので注意しましょう。

3.3.3 RAID2

最低 5 台の HDD が必要。

最小構成の場合、実際にデータを保存する HDD を 2 台と、誤り訂正符号 (ハミング符号) を入れておく HDD が 3 台になります。長所は、読み書き速度が極めて高速であることと、極めて優れた耐障害性 (全 RAID の中で最強) を持つことです。

ハミング符号を用いてデータ修復をしなければならないほど HDD の信頼性は低くない上に、ディスクの使用効率が恐ろしく悪いので実用性は無く、今も昔もほとんど使われていません。

3.3.4 RAID3

最低 3 台の HDD が必要。

n 台の HDD を使う場合、*n* – 1 台の HDD でストライピング (RAID0) を行い、残りの 1 台に誤り訂正符号 (パリティ) を入れておきます。RAID2 とは違って、誤り訂正符号は排他的論理和によるものです。

長所は、RAID2 と違い誤り訂正符号に排他的論理和を使っているので、読み書きがそれなりに高速であることと、RAID1,2 に比べてディスクの使用効率が良いことです。短所としては、ストライピングをしている HDD は全て同期を取らなくてはならないのでそのためのコストがかかること、およびパリティドライブの転送速度がボトルネックとなることなどです。また、アクセス単位がビットやバイトなので、小さなデータをちまちま書き込むのには向いていません。

ビデオ編集機器などではよく用いられた RAID ですが、現在は RAID5 に取って代わられました。

3.3.5 RAID4

最低 3 台の HDD が必要。

構成は RAID3 とほぼ同じですが、アクセス単位がビットやバイトではなくブロックと呼ばれるもっと大きなデータの塊になりました。これによりストライピングをしている HDD は同期を取らなくてもよくなったので、RAID3 の複雑さは解消されました。また、読み書き速度がさらに向上しました。

それでも相変わらずパリティドライブの転送速度がボトルネックになること、そして次に述べる RAID5 に性能面で劣るがために、廃れました。

3.3.6 RAID5

最低 3 台の HDD が必要。

RAID3、4 では 1 台の HDD に集中していたパリティを複数の HDD に分散して書き込むことによって、パリティドライブ (と呼ばれたもの) の転送速度がボトルネックにならなくなりました。

欠点としては、HDD のうちの 1 台が壊れたときの復旧作業がそれなりに複雑であることと、書き込み時にパリティの照合を行うため書き込み速度が少々低速であることです。また、同時に 2 台の HDD が壊れてしまうと、復旧ができなくなります。

3.3.7 RAID6

最低 4 台の HDD が必要。

RAID5 ではパリティを複数の HDD に分散させて置いていました。RAID6 ではこのパリティを 2 重に用意し、それを従来のパリティの位置より微妙にずらして置いておくことで、同時に 2 台の HDD が壊れても復旧可能になりました。

欠点としては、パリティを 2 つ作るため RAID5 よりも速度が劣ることです。とは言っても最近の HDD は高速なので、特に気にはならないでしょう。

3.3.8 RAID7™

この RAID だけ見慣れない ™ マークがついていますね。なぜかというと、RAID7 は Storage Computer Corporation の登録商標だからです。うん、この RAID

を触る機会は一生無いでしょう。というわけで省略[14]。

以上より、やるなら RAID5 が良いかな？ 金銭的に余裕があるなら RAID6 をやっても良いでしょう。あるいは、どこぞの絵描きさんみたいに RAID0 と 1 を組み合わせて RAID10！ なんてことをやるのも良いかもしれません。

なお、HDD を冗長化しないとなると、マシンそのものを冗長化することになるでしょうね。つまり同じマシンを 2 台用意しておく。う〜ん、俺には HDD のみの冗長化の方が経済的に見えます。って、そんなことを言っているからマシンがぶっ壊れたときに何も出来なくなるんですね！ 気をつけましょう。

3.4 場所

結局、買ったマシンやら HDD やらは全部自分の部屋か、あるいは会社や研究室に置くことになるのでしょう。しかし、それでは天変地異に耐えることが出来ません。あなたは生き延びても、データの方が物理的に死んでしまう可能性があります。あなたとデータとどっちの方が価値があるか、という重大な問題に関しては敢えて議論しません。ここではデータの方が価値があるとして話を進めます。

企業とかの場合、遠隔地にデータセンターを設けるのが常套手段です。どれくらい遠くに設けるかですが、近すぎると天変地異の巻き添えを食らうかもしれず、しかし遠すぎるとメンテナンスをするのが大変です。経験的には、本部から 60km ほど離れたところにデータセンターがあると良いと言われています。

ですが、我々一般庶民がデータセンターを建てるなんてできるはずがありませんね。そこで、俺は次の方法をとっています。

3.4.1 Sourceforge の SVN Repository を使う

Sourceforge は、オープンソースのプロジェクトを支援するためのサイト、およびツール群です。実態はよくわからないんですが、Sourceforge 側が間違ってデータを消す、ということはほぼ確実にないと思います。世界中にミラーがあり、それこそ国が 10 くらい滅びそうなほど巨大な隕石が地球を直撃してもデータは生き延びるでしょう。俺は未踏ユースのプロジェクトをここで管理しています。ここで管理していたプロジェクトは全て生き延びました。ここで管理していなかったプロジェクトは、メイン PC が壊れるのとともに消え去りました。

3.4.2 P2P による共有ソフトで配信

これは流石にやっていませんが、地味に相当有効な手段なのではないかと最近思い始めました。なぜなら、世界中の全てのマシンに俺のプロジェクトの一部が入っていれば、それこそ反重力爆弾[15]が暴発しても、世の中にマシンが 1 台でもあれば (部分的ではあるが) 俺のプロジェクトは生き延びるからです。もっとも、例えば会社の極秘情報とかはネットに流すわけにも行かないので、この方法が現実的になる日はあまり近くなさそうです。

4 所感

まぁ、ここまでいろいろ書いてきたわけですが、一番大事なことについて書くのを忘れていました。データを護る方法には様々なものがあります。どれもこれも、少なからず金がかかるものです。だから、俺のデータは絶対に消えないようにあーだこーだと言うのも良いんですが、それより前に是非一度、自分の持っているデータの価値について考えていただきたいのでです。恐ろしい額の金を払って、ゴミ以下のデータを護る必要なんて無いですからね。

そして、もう 1 つ。世の中には、巨額をかけてでも護らなければならないくらい価値のあるデータは存在します。それは最初からこの世にあったものではなくて、誰かが作ったものです。読者にも、それくらい価値のあるデータを作っていただきたい、そしてそういうデータはなんとしてでも護っていただきたいです。

データを与えられるだけでなく、データを作れる人になってください。

[14] というよりも資料が見つからんかった。すんません。

[15] 「鉄腕アトム」に出てきた、星 1 つなら簡単にぶっ壊せるくらい強力な爆弾。